XOJO

Web-Tutorial

2014 Release 1

Xojo, Inc.

Kapitel 1

Einführung

In diesem Tutorial lernen Sie die Erstellung einer Web-App.

Abschnitt 1

Über dieses Web-Tutorial

XOJO STARTEN

1. **Laden Sie die Installationsdatei für Ihr Betriebssystem hier herunter: http://www.xojo.com/download**
2. **Führen Sie das Installationsprogramm aus.**
3. **Starten Sie Xojo.**
4. **Wählen Sie in der Projektauswahl Web aus und klicken Sie auf OK.**

Dieses Tutorial wendet sich an Xojo- und Programmier-Neulinge. Es stellt eine Einführung in die Web-Entwicklungsumgebung dar und führt Sie durch die Erstellung einer echten Web-Anwendung.

Die Bearbeitungszeit für dieses Tutorial liegt bei etwa einer Stunde.

Aufgemerkt: Wenn Sie bereits über Erfahrungen mit anderen Programmiersprachen verfügen, sollten Sie sich auch das Benutzerhandbuch und die Sprachreferenz anschauen.

Copyright

Alle Inhalte copyright 2014 Xojo, Inc. Alle Rechte vorbehalten. Kein Teil dieser Dokumentation oder der damit verbundenen Dateien darf ohne vorheriges schriftliches Einverständnis des Herausgebers reproduziert, kopiert oder versandt werden, was elektronische, photomechanische oder andere Aufzeichnungs- oder Übermittlungsverfahren einschließt.

Warenmarke

Xojo ist eine eingetragene Warenmarke von Xojo, Inc.

Dieses Buch benennt Produktbezeichnungen und Dienstleistungen, die als registrierte und unregistrierte Warenmarken oder Service Marks Ihrer Rechteinhaber bekannt sind. In diesen Buch finden sie ausschließlich zu Lehrzwecken Erwähnung.

Begriffe, die vermutlich als Warenzeichen oder Service Mark eingetragen sind, werden in Großschreibung wiedergegeben, obwohl Xojo, Inc. über ihren Markenzustand keine verbindliche Zusage treffen kann. Die Verwendung dieser Begriffe stellt keine Verletzung der zugehörigen Marken- oder Urheberrechte dar. Xojo, Inc. steht in keiner Verbindung zu den in diesem Buch erwähnten Produkten oder ihren Herstellern.

Anmerkung des Übersetzers:
Zurzeit liegt Xojo nicht lokalisiert vor, das heißt die Programmumgebung erscheint in englischer Sprache. Deshalb werden in diesem Tutorial die englisch erscheinenden Begriffe auch englisch wiedergegeben, bei ihrer Ersterwähnung aber aus Verständnisgründen deutsch übersetzt. Sie können sich über das Glossar dieses Tutorials die Übersetzungen der Begriffe anzeigen lassen.

Abschnitt 2

Bedeutung von Abkürzungen, Zeichen und Schriften

Dieses Tutorial verwendet Screenshots der Xojo-Windows-, OS X- und Linux-Versionen. Aufbau des Benutzerinterfaces und Funktionalität sind auf allen Plattformen identisch. Abweichungen in der Darstellung ergeben sich aus den unterschiedlichen graphischen Benutzerinterfaces der Betriebssysteme.

- **Fettschrift** wird sowohl zur Markierung der Einführung eines neuen Begriffs verwendet als auch zur Hervorhebung wichtiger Konzepte.
- Buchtitel wie *Xojo-Benutzerhandbuch* sind kursiv geschrieben.
- Angaben zur Verwendung eines Eintrags aus den Programm-Menüs werden in der Form „Wählen Sie Datei ↠ Neues Projekt“ gegeben. Dieses Beispiel ist die Kurzform von „Wählen Sie ‚Neues Projekt‘ aus dem Datei-Menü.“
- Tastaturkürzel bestehen aus einer Reihe von Tasten, die in der angegebenen Reihenfolge gedrückt und gehalten werden sollten. Unter Windows und Linux wird die STRG-Taste als Steuerungstaste verwendet, unter OS X dient dazu die ⌘ (Befehl)-Taste. Das Kürzel „STRG+O“ oder „⌘-O“ bedeutet also, die STRG-Taste auf Windows oder Linux-Rechnern oder die ⌘-Taste auf OS X gedrückt zu halten und dann zusätzlich die „O“-Taste zu drücken. Heben Sie den Finger von der STRG- bzw. Befehl-Taste erst, wenn Sie die „O“-Taste nicht mehr drücken.
- In Anführungszeichen stehen Begriffe, die Sie eingeben sollten, wie „StartKnopf”.
- Mitunter wird von Ihnen erwartet, Programmbefehle in den Code-Editor einzugeben. Diese werden in grauen Kästen angezeigt:

```
ShowURL(GewählteURL.Text)
```

Bitte beachten Sie bei der Codeeingabe folgendes:

- Geben Sie jede vorgegebene Zeile als einzelne Zeile in den Code-Editor ein. Versuchen Sie nicht, mehrere Programmzeilen zu einem Satz zusammenzufassen oder eine lange Zeile mittels Zeilenschaltungen (Return) zu unterteilen.
- Fügen Sie keine weiteren Leerzeichen ein, wenn in den Programmierzeilen im Tutorial keine stehen.
- Sie können die Codezeilen natürlich auch über die Zwischenablage aus dem Tutorial in den Code-Editor kopieren.

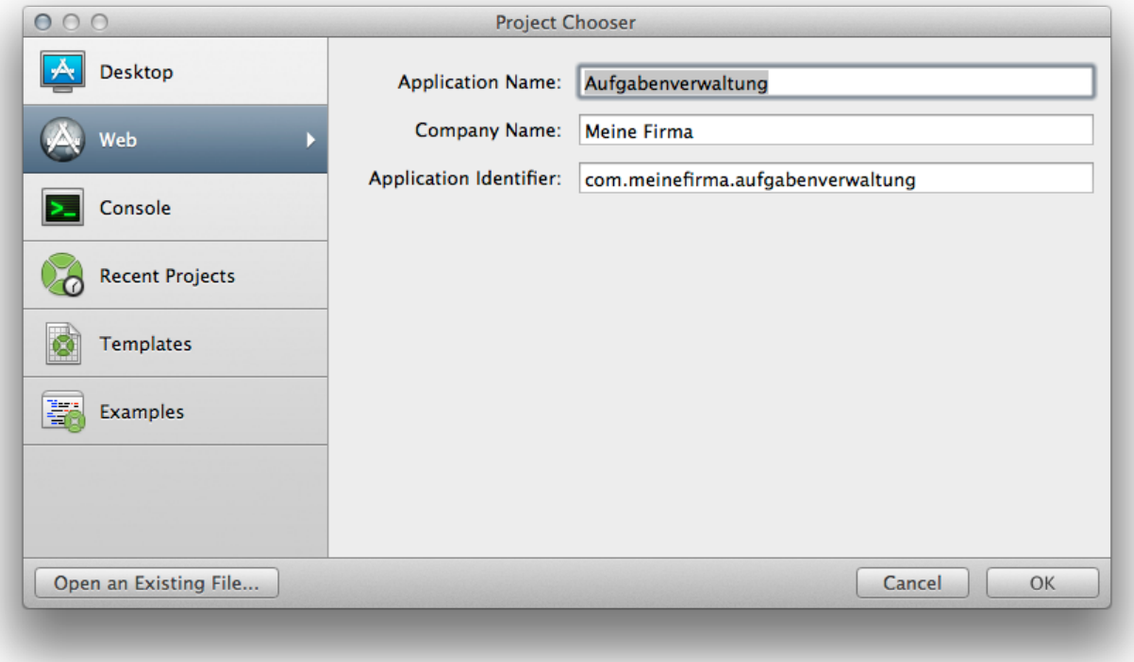
Wenn Sie Ihre Anwendung kompilieren, überprüft Xojo zunächst Ihren Code auf Tipp- und Syntaxfehler. Sollte dabei ein Fehler erkannt werden, erscheint eine Fehlerliste zur Überprüfung am Fuß des Xojo-Hauptfensters.

Erste Schritte

Falls Sie das noch nicht getan haben: Starten Sie Xojo.

Doppelklicken Sie das Xojo-Programmsymbol, um Xojo zu starten. Nach Abschluss des Ladevorgangs erscheint das Projektauswahl-Fenster.

Abb. 1.1 Das Projektauswahl-Fenster (Project Chooser)



Xojo erlaubt die Erzeugung dreier verschiedener Anwendungsarten (Desktop, Web und Kommandozeile).

In diesem Tutorial erstellen wir eine Web-Applikation. Klicken Sie also auf Web.

Sie sehen nun drei Texteingabefelder: Anwendungsname (Application Name), Firmenname (Company Name) und Anwendungs-ID (Application Identifier).

Application Name ist der Name Ihrer Anwendung. Der hier eingegebene Begriff bestimmt den Dateinamen Ihrer Anwendung.

Company Name ist der Name Ihres Unternehmens. Sie können dieses Feld auch leer lassen.

Application Identifier stellt eine eindeutige Kennung Ihrer Anwendung dar. Aus den Eingaben der Felder Application Name und Company Name wird eine Vorgabe für dieses Feld erzeugt. Sie können sie aber beliebig verändern.

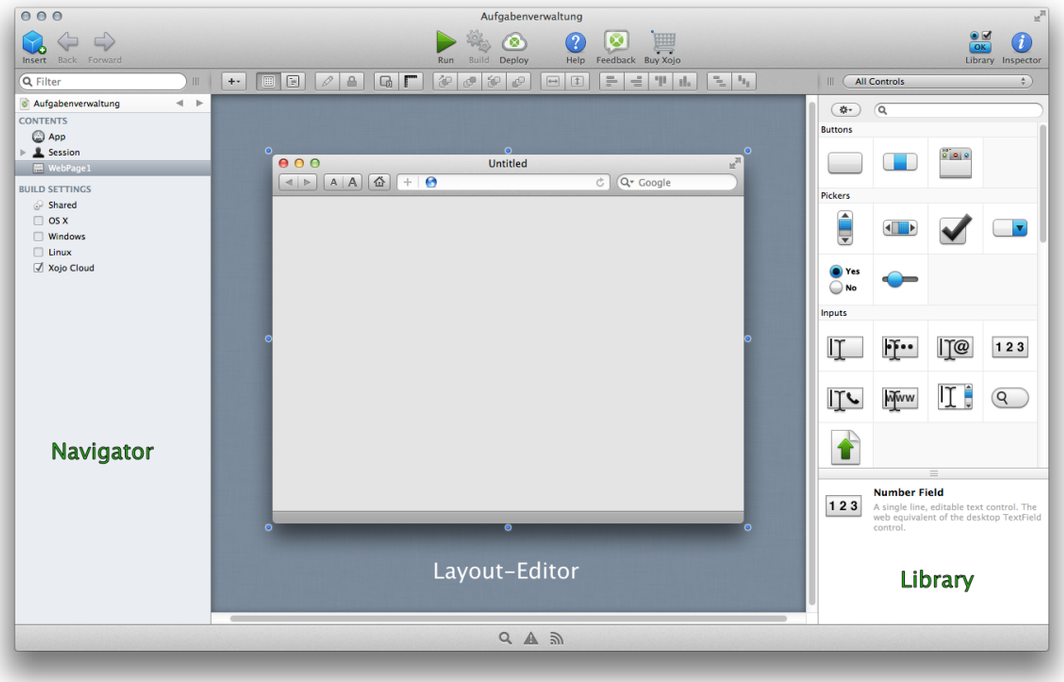
Geben Sie „Aufgabenverwaltung“ als Anwendungsnamen ein und verändern Sie die Vorgabe für den Company Name ganz nach Lust und Laune oder lassen Sie sie so wie vorgegeben.

Klicken Sie auf **OK**, um das Hauptfenster von Xojo zu öffnen – den Arbeitsbereich / Workspace –, in dem Sie Ihre Anwendung erstellen.

Der Arbeitsbereich (Workspace)

Xojo öffnet den Arbeitsbereich (**Workspace**) mit den für Ihre Anwendungsart passenden Vorgabewerten im **Navigator** und im **Layout-Editor**.

Abb. 1.2 Das Xojo Hauptfenster



Navigator: Das Panel am linken Bildschirmrand listet alle Bestandteile Ihres Projekts auf. Standardmäßig sehen Sie hier WebPage1 (die ausgewählt ist), das App- und das Session-Objekt. Mittels des Navigators bewegen Sie sich innerhalb Ihres Projekts.

Layout-Editor: Das mittlere Gebiet des Workspace ist der Layout-Editor. Mit diesem erstellen Sie das Benutzerinterface für die Fenster Ihrer Anwendung. Der Layout-Editor liefert eine Vorschau auf das Erscheinungsbild Ihres Programmfensters. In der Abbildung ist das Fenster leer, weil noch keine Steuerelemente aus der Bibliothek (**Library**) hinzugefügt wurden.

Bibliothek (**Library)**: Das Gebiet am rechten Rand des Hauptfensters stellt die Library dar, in der Sie die Steuer- und Interface-Elemente finden, die Sie zu einem Fenster oder zum Projekt hinzufügen können. Sie gestalten Ihr Fenster, indem Sie die Elemente aus der Library auf dieses hinüberziehen oder indem Sie auf ein Element in der Library doppelklicken.
Sie können das Erscheinungsbild der Elemente in der Library verändern, indem Sie auf das kleine Zahnradsymbol klicken und eine andere Einstellung auswählen.

Aufgemerkt: Sollte die Library nicht sichtbar sein, klicken Sie auf das Library-Symbol in der Werkzeugleiste.

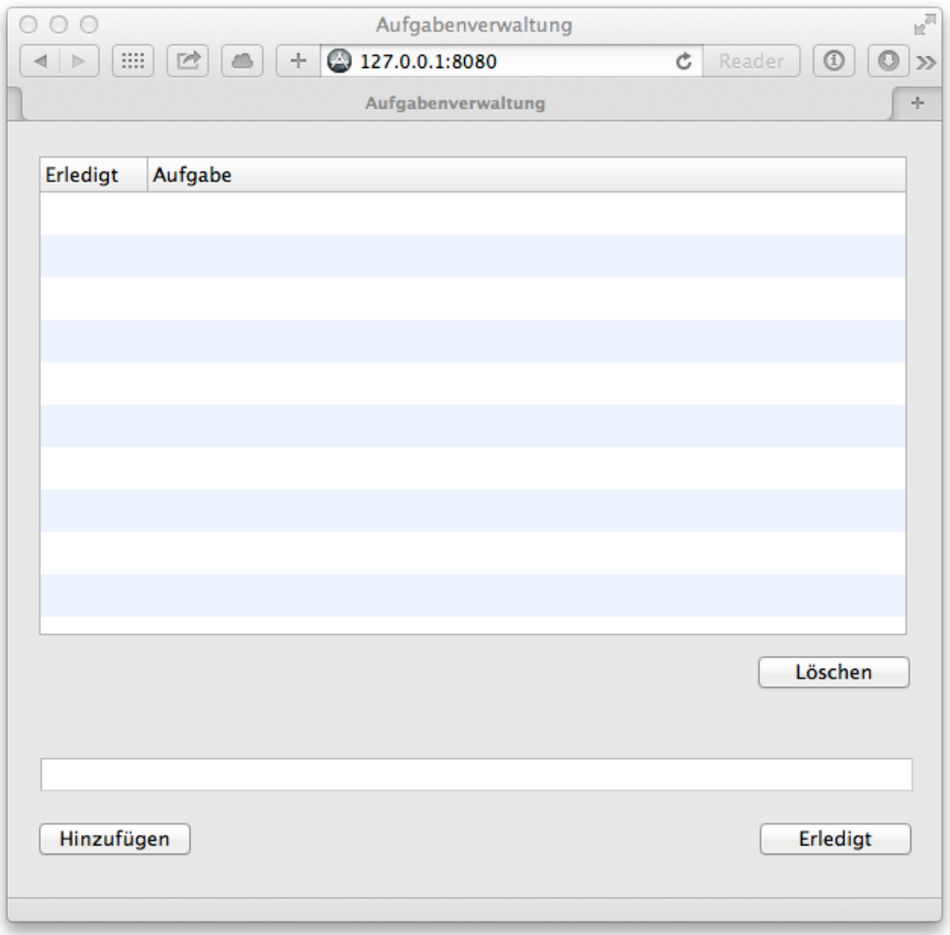
Inspector: In der Abbildung nicht sichtbar ist der Inspector, in dem Sie die Eigenschaften des ausgewählten Steuerelements sehen und verändern können. Der Inspector teilt sich das rechte Panel des Hauptfensters mit der Library.
Sie können sich den Inspector anzeigen lassen, indem Sie auf das Inspector-Symbol in der Werkzeugleiste klicken.
Der Inspector liefert Informationen über das im Navigator oder Editor ausgewählte Element. Das Aussehen des Inspectors wird dem gerade gewählten Element angepasst.
Sie verändern eine Eigenschaft des angezeigten Elements, indem Sie einen Wert in das Texteingabefeld rechts der Feldbezeichnung eingeben.

Die Anwendung

Aufgabenverwaltung

Wir werden in diesem Tutorial eine Anwendung erstellen, mit der Sie beliebige Aufgaben wie folgt verwalten können:

Abb. 1.3 Aufgabenverwaltung als Web-Anwendung



Sie geben Aufgaben in das Textfeld ein und fügen Sie mittels Klick der Liste hinzu. Angelegte Aufgaben können per Klick ausgewählt und gelöscht oder als abgeschlossen markiert werden.

Die Aufgabenverwaltung verwendet drei Steuerelemente:

WebTextField: Ein **WebTextField**-Steuerelement nimmt Texteingaben entgegen. In diesem Projekt wird die Aufgabe in ein Webtextfield am Fuße des Fensters eingegeben.

WebButton: Mit einem **WebButton** kann eine Aktion ausgelöst werden. In diesem Projekt verwenden wir verschiedene Buttons für unterschiedliche Aufgaben.

WebListBox: Eine **WebListBox** kann eine Reihe von Daten tabellarisch anzeigen. In diesem Projekt wird eine Weblistbox verwendet, um die erfassten Aufgaben aufzulisten.

In den folgenden Kapiteln gehen wir Schritt für Schritt durch die Erstellung der Anwendung vom Design der Benutzerschnittstellen bis zum Hinzufügen des Programmcodes.

Kapitel 2

Gestaltung des Benutzerinterface

Wir entwerfen die Benutzerschnittstelle für die Aufgabenverwaltung.

Aufgabenliste

Hinzufügen der Aufgabenliste

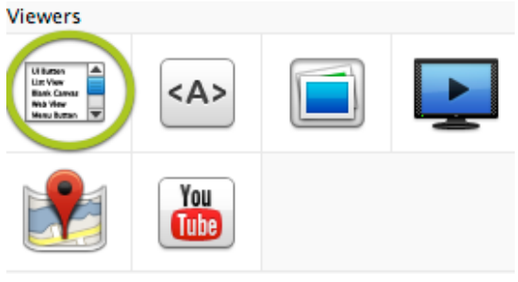
Sie sollten Xojo mit geöffnetem Layout-Editor vor sich haben. Falls nicht, lesen Sie bitte Kapitel 1, Abschnitte 3 und 4.

Wir werden der Webseite jetzt eine ListBox hinzufügen. In dieser sollen später die Aufgaben angezeigt werden.

1. Hinzufügen der Listbox

   Klicken Sie in der Library auf das Symbol der ListBox und ziehen Sie sie in die obere linke Ecke des Layout-Editors.

   Abb. 2.1 ListBox-Steuerelement

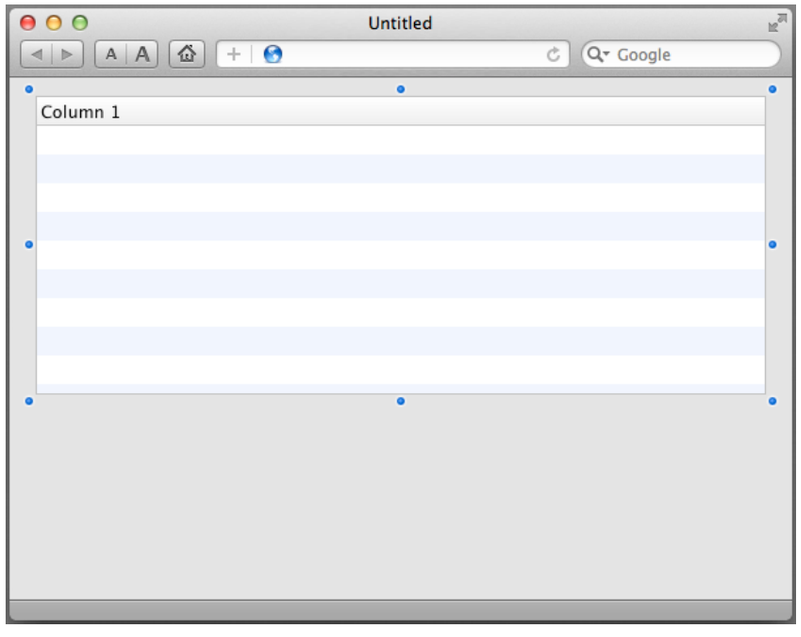
   Wenn Sie den Fensterrändern nahekommen, werden Sie Positionierungshilfslinien bemerken, die Ihnen bei der Platzierung Ihres Elements helfen. Lassen Sie die Maustaste los, wenn Ihnen die Position der ListBox innerhalb des Fensters gefällt.

2. Anpassung der ListBox-Größe:

   Klicken Sie auf die ListBox, damit ihre Anfasser erscheinen. Klicken Sie auf den Anfasser in der rechten unteren Ecke und ziehen Sie bei gedrückter Maustaste, bis die ListBox etwa ⅔ des Fensters ausfüllt.

3. Ihre Website sollte nun ungefähr so aussehen:

   Abb. 2.2 Webseiten-Layout mit ListBox

Schaltflächen (Buttons)

Hinzufügen der Buttons

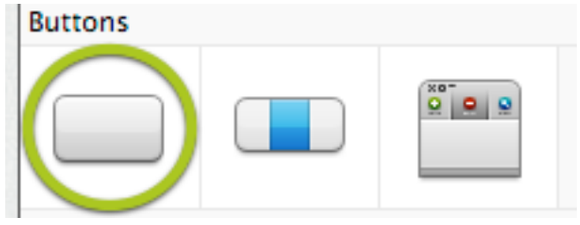
Wir werden jetzt die drei Schaltflächen hinzufügen, die die Aufgabenverwaltung benötigt:

Der „Löschen“-Button löscht Aufgaben von der ListBox, der „Hinzufügen“-Knopf fügt Aufgaben hinzu und die „Erledigt“-Schaltfläche markiert Aufgaben als abgeschlossen.

1. Hinzufügen der Löschen-Schaltfläche:

   Klicken Sie in der Library auf das Button-Steuerelement und ziehen Sie es auf die Webseite unter die untere rechte Ecke der ListBox.

Abb. 2.3 Button-Steuerelement



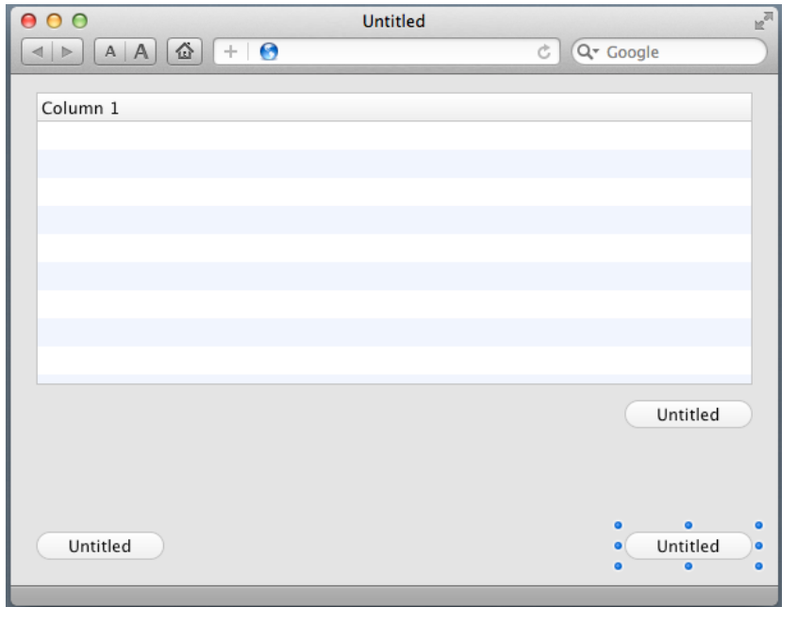
   Achten Sie auf die automatischen Hilfslinien, um den Button bündig mit der rechten Kante der ListBox zu positionieren.

2. Hinzufügen des Hinzufügen-Buttons:
   Klicken Sie in der Library auf das Button-Steuerelement und ziehen Sie es in die Nähe der linken unteren Fensterecke. Die Hilfslinien helfen Ihnen wieder dabei.

3. Hinzufügen des „Erledigt“-Buttons:
   Ziehen Sie wiederum ein Button-Steuerelement in die rechte untere Ecke des Fensters.

4. Ihre Webseite sollte jetzt ungefähr so aussehen:

Abb. 2.4 Webseiten-Layout mit platzierten Knöpfen



Das Textfeld (TextField)

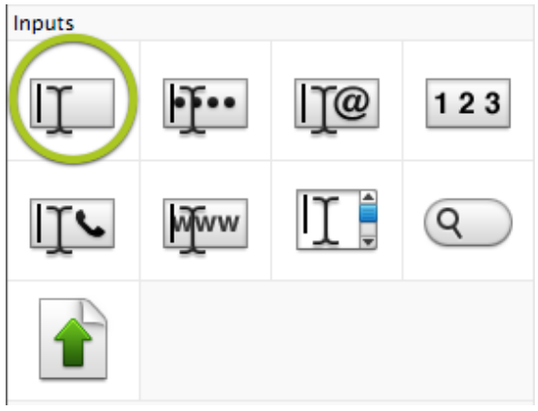
Hinzufügen des Textfelds

Im Textfeld gibt der Benutzer neue Aufgaben ein.

1. Klicken Sie in der Library auf ein Textfeld (TextField) und ziehen Sie es in das Fenster zwischen Löschen- und Hinzufügen-Knopf.

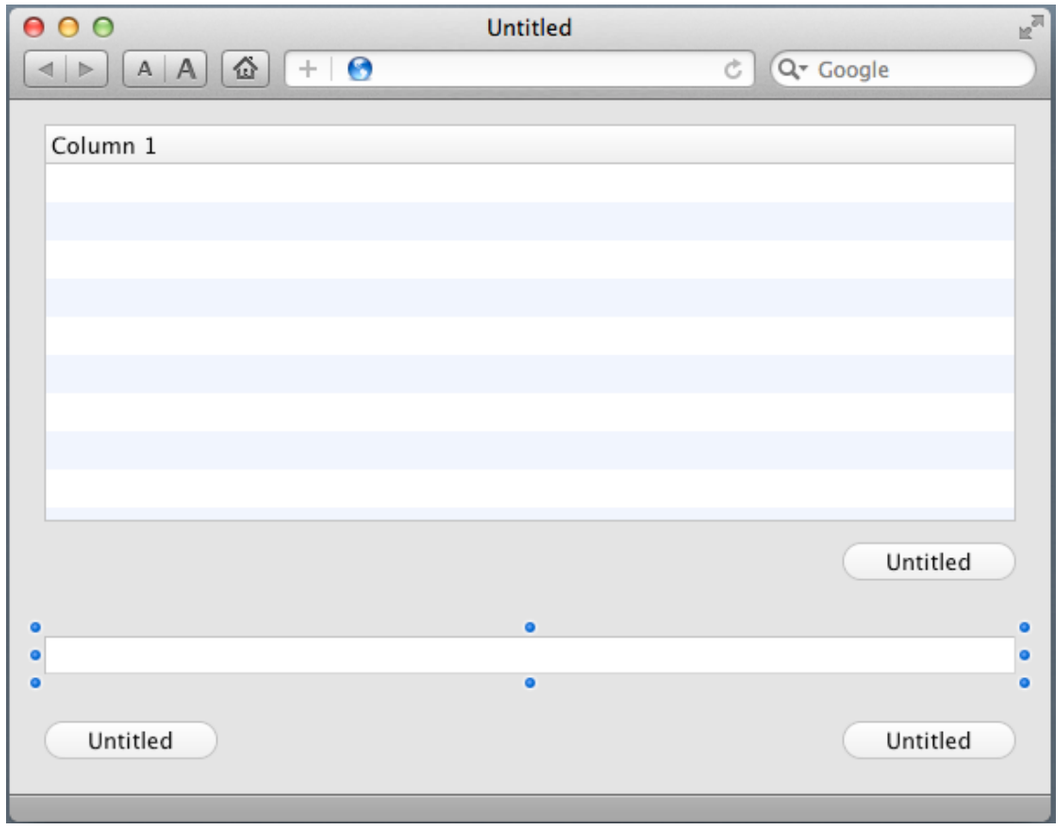
2. Verändern Sie die Größe des Felds: Ziehen Sie es mittels seiner Anfasser und unterstützt von den intelligenten Hilfslinien auf die Breite der ListBox.

Abb. 2.5 TextField-Steuerelement



3. Ihre Webseite sollte jetzt in ungefähr so ausschauen:

Abb. 2.6 Vollständiges Webseitenlayout



Eigenschaften (Properties)

Was ist eine Eigenschaft (Property)?

Eine **Property** ist ein Wert einer Klasse, wie etwa die Breite eines Fensters. Mittels Veränderung der Properties einer **Klasse** können Sie Verhalten und Aussehen einer **Instanz** der Klasse verändern.

In diesem Projekt werden Sie verschiedene Properties der Web-App und ihrer Steuerelemente verändern. Dazu gehören:

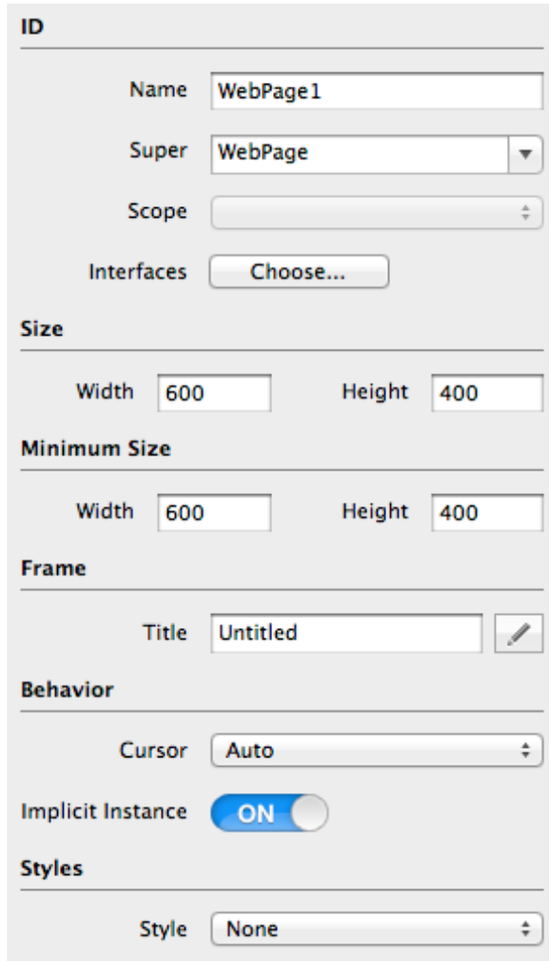
- Umbenennung aller Steuerelemente (und der Webseite), damit ihre Namen ihre Funktionalität besser beschreiben und sie somit eindeutiger innerhalb des Programmcodes ansprechbar sind.
- Verändern der Schaltflächen-Titel.
- Setzen von Verriegelungs-Properties, um sicherzustellen, dass die Steuerelemente mit Größenveränderung des Fensters korrekt skaliert werden.

Inspector

Im Inspector werden die Fenster- und Steuerelement-Properties verändert. Der Inspector teilt sich das rechte Panel des Hauptfensters mit der Library. Sie können sich den Inspector anzeigen lassen, indem Sie auf das Inspector-Symbol in der Werkzeugleiste klicken.

Sie können mittels der Tastaturkürzel ⌘-L auf OS X (STRG-K unter Windows und Linux) zur Library und ⌘-I auf OS X (STRG-I unter Windows und Linux) zum Inspector wechseln.

Abb. 2.7 Webseiten-Properties im Inspector



Abschnitt 5

Webseiten-Properties

Wenn Sie dies noch nicht getan haben sollten, rufen Sie jetzt den Inspector durch Klick auf das Inspector-Symbol in der Werkzeugleiste auf.

Sie sollten nun die Properties **Name** und **Title** der Webseite ändern:

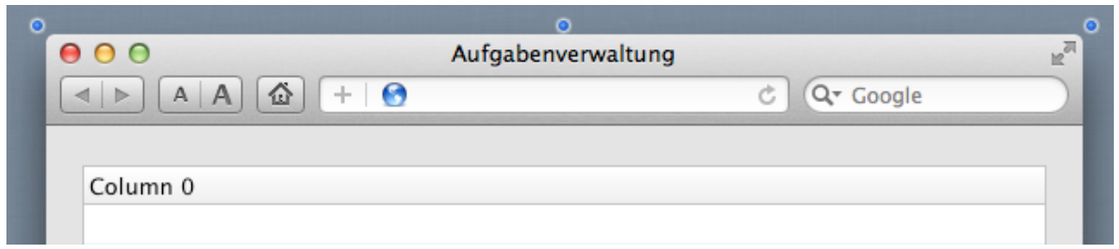
1. Klicken Sie im Layout-Editor auf die Titelzeile des Webseite, um diese auszuwählen. Der Inspector zeigt jetzt die Properties der Webseite an.

2. Ändern Sie den Text im Feld Name (ganz oben im Inspector in der Gruppe ID) von „WebPage1“ in „AufgabenverwaltungsSeite“. Drücken Sie den *Zeilenschalter* (*Return*), um die Namensänderung im Navigator zu sehen.

3. Ändern Sie den Text im Feld Title (weiter unten in der Frame-Gruppe) von „Untitled“ zu „Aufgabenverwaltung“.
   Nach Eingabe von *Return* sehen Sie den neuen Namen in der Titelzeile der Webseite:

Abb. 2.8 Namensänderung im Navigator

Abb. 2.9 Webseiten-Layout mit neuem Titel



Abschnitt 6

ListBox-Eigenschaften

In der Listbox sollen die vom Benutzer eingegebenen Aufgaben aufgelistet und gespeichert werden. Sie müssen die folgenden Properties verändern: **Name, ColumnCount, Default Value, ColumnWidths** und **Locking.**

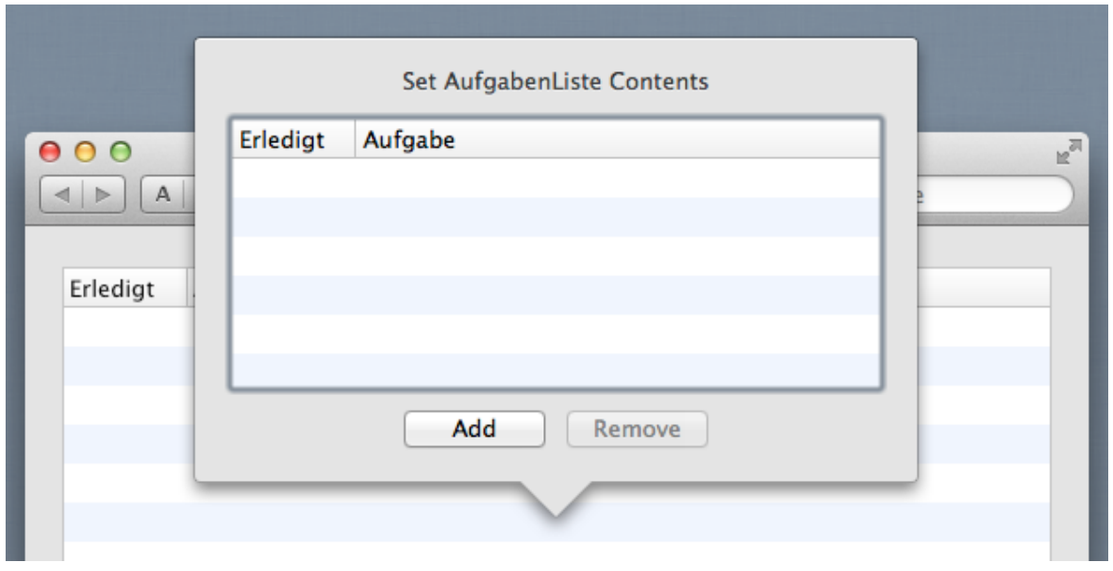
1. Klicken Sie zunächst im Layout-Editor die ListBox an, um sie auszuwählen. Der Inspector zeigt nun die Properties der Listbox an.

2. Ändern Sie im Feld Name (in der ID Gruppe) den Text von „Listbox1“ in „AufgabenListe“. Drücken Sie *Return*, um die Veränderung in den Navigator zu übernehmen.

3. Die ListBox soll zwei Spalten haben, eine für den Abschluss-Status ein eine für den Namen der Aufgabe. Ändern Sie daher den Wert im Feld **ColumnCount** von „1“ auf „2“. Drücken Sie *Return*, um die ListBox mit zwei Spalten im Webseitenlayout angezeigt zu bekommen.

4. Wir werden nun die Kopfzeile ändern, um die Spalten mit aussagekräftigen Namen zu versehen. Klicken Sie auf den „Set Default Value of AufgabenListe...“-Knopf (Vorgabewerte einstellen) in der Layout-Editor-Werkzeugleister ( ). Damit öffnen Sie das PopOut-Fenster für den Werte-Editor:

Abb. 2.10 Vorgabewerte für die Aufgabenliste



   a. Klicken Sie auf „Column 0“ (Spalte 0) in der Kopfzeile, um diesen Wert auszuwählen, und klicken Sie erneut darauf, um ihn zu ändern. Geben Sie „Erledigt“ ein und drücken Sie *Return*.

b. Klicken Sie 2 x auf „Column 1“in der Kopfzeileund ändern Sie den Namen in „Aufgabe“. Drücken Sie *Return*.

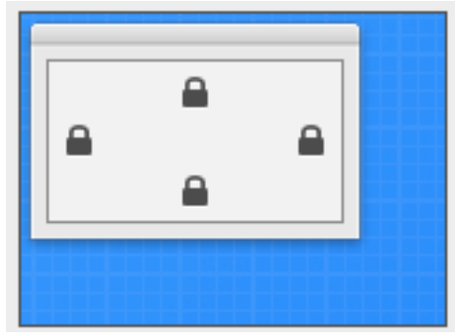
c. Klicken Sie den „Set Default Value of AufgabenListe...“-Button erneut, um das Eingabefenster zu schließen. Die neuen Titel erscheinen in der Webseite des Layout-Editors.

5. Da die Spalte „Erledigt“ nur eine Checkbox erhalten wird, muss sie nicht so breit sein. Ändern Sie den Wert im Feld **ColumnWidths** (Spaltenbreiten)von „*“ zu „70,*“. Drücken Sie *Return*, um die Änderungen zu sehen.

   Die Angabe von „70,*“ sagt der ListBox, dass die erste Spalte eine Breite von 70 Pixeln haben sollte und sich die anderen Spalten den Rest der verfügbaren Breite teilen.

6. Zuletzt müssen noch die **Verriegelungspositionen** verändert werden, damit die Listbox bei Größenveränderungen des Fensters mitskaliert wird.
   Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens, aber geöffnete an der unteren und der rechten Seite.
   Klicken Sie auf die Schloss-Symbole, bis alle vier Symbole als verriegelt angezeigt werden.

Abb. 2.11 Locking für die AufgabenListe



Schaltflächeneigenschaften

Die drei auf dem Fenster platzierten Buttons sollen Aktionen auslösen. Sie sollten die folgenden Properties jedes Buttons ändern: **Name, Caption** und **Locking.**

Die Löschen-Schaltfläche

Der Button Löschen wird verwendet, um Aufgaben von der AufgabenListe zu löschen.

1. Klicken Sie im Layout-Editor zunächst auf den Löschen-Button, um ihn auszuwählen (das ist der Button direkt unterhalb der Listbox). Der Inspector zeigt nun die Properties des Löschen-Buttons an.

2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „WebButton1“ in „LöschButton“. Drücken Sie *Return*, um die Veränderung in den Navigator zu übernehmen.

3. Geben Sie im Feld **Caption** (= Titel, Beschriftung) (unten im Inspector in der Gruppe Appearance (Erscheinungsbild)) statt „Untitled“ „Löschen“ ein. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Button des Fensters zu sehen.

4. Nun sollten Sie das Locking ändern, damit der Lösch-Button bei Größenveränderungen des Fensters stets auf der rechten Seite unterhalb der Listbox bleibt.

   Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite.
   Klicken Sie auf die Symbole, bis oberes und linkes Symbol geöffnet sowie rechtes und unteres Symbol verriegelt sind.

Abb. 2.12 Locking für den LöschButton

Die Hinzufügen-Schaltfläche

Der Button „Hinzufügen“ wird verwendet, um in das TextField eingegebene Aufgaben zur Aufgabenliste hinzuzufügen.

1. Klicken Sie im Layout-Editor auf den Hinzufügen-Button, um ihn auszuwählen (dies ist die Schaltfläche in der linken unteren Ecke des Fensters unterhalb des TextFields). Der Inspector zeigt nun die Properties des WebButtons an.

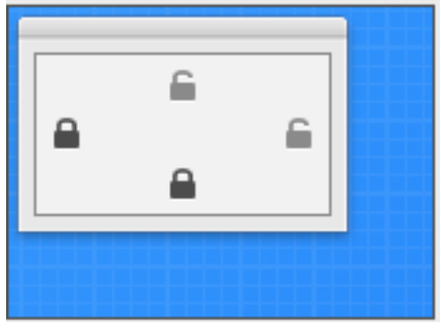
2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „WebButton2“ in „HinzufügenButton“. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Navigator zu sehen.

3. Geben Sie im Feld Caption (unten im Inspector in der Gruppe Appearance) statt „Untitled“ „Hinzufügen“ ein. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Button des Fensters zu sehen. Ziehen Sie den Button im Layout-Editor ggf. ein bisschen breiter, um den ganzen Text anzuzeigen.

4. Jetzt sollten Sie die Verriegelungspositionen des Buttons überprüfen, damit die Schaltfläche bei Größenveränderungen des Fensters immer in der linken unteren Ecke bleibt.

Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite.
Klicken Sie auf die Symbole, bis oberes und rechtes Symbol geöffnet sowie linkes und unteres Symbol verriegelt sind.

Abb. 2.13 Locking für den HinzufügenButton



Der Erledigt-Knopf

Ein Klick auf diese Schaltfläche markiert die angewählte Aufgabe als erledigt.

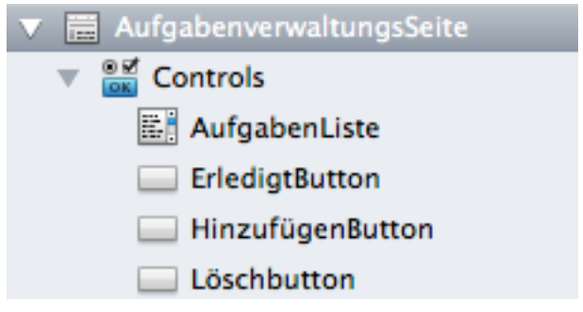
1. Klicken Sie im Layout-Editor auf den Anzeigen-Button, um ihn auszuwählen (dies ist die Schaltfläche in der rechten unteren Ecke des Fensters unterhalb des TextFields). Der Inspector zeigt nun die Properties des Pushbuttons an.

2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „WebButton3“ in „ErledigtButton“. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Navigator zu sehen.

3. Geben Sie im Feld Caption (unten im Inspector in der Gruppe Appearance) statt „Untitled“ „Erledigt“ ein. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Button des Fensters zu sehen.

In der Projektliste sollten die neuen Steuerelemente der AufgabenverwaltungsSeite nun so angezeigt werden:

Abb. 2.14 Steuerelemente nach Umbenennung

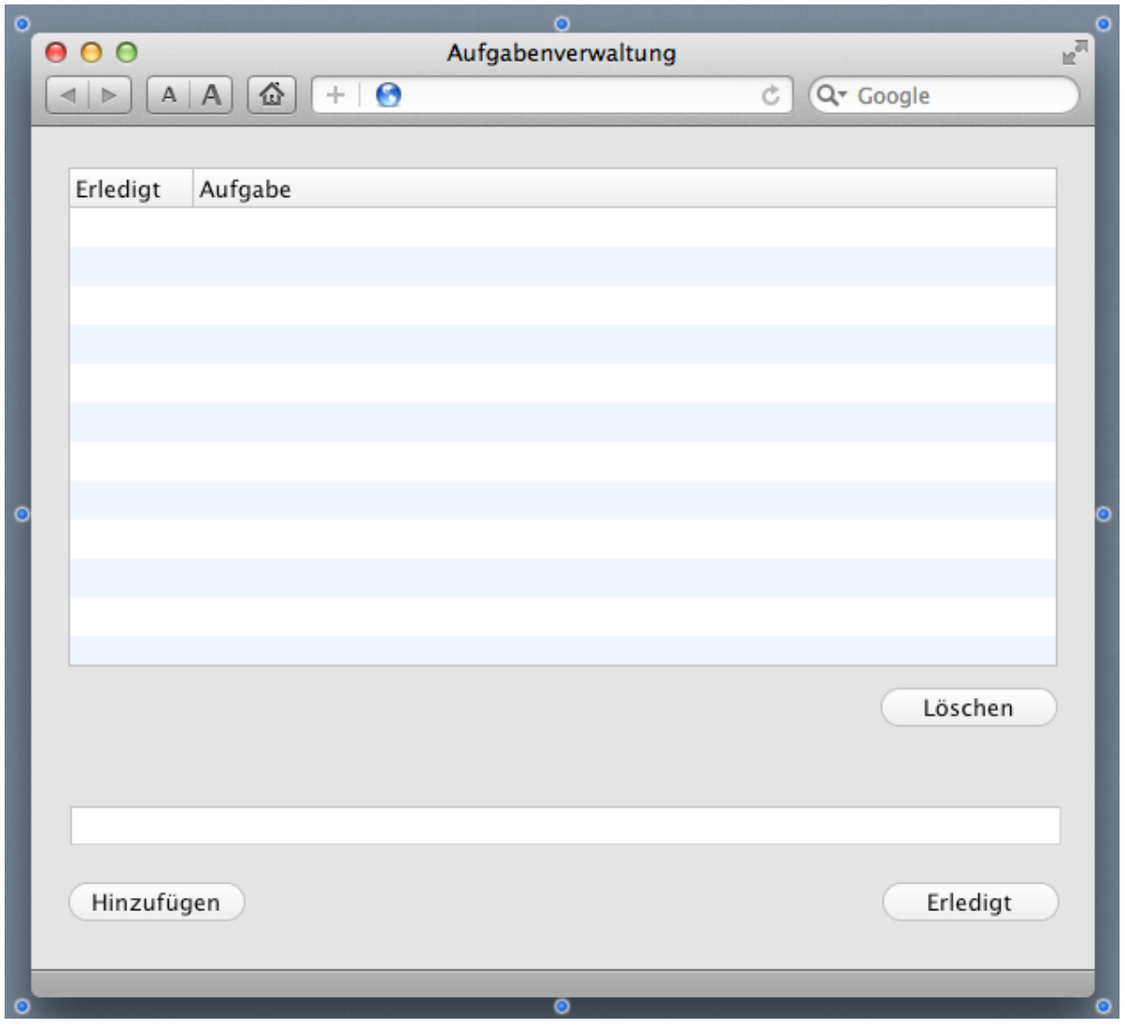


4. Jetzt sollten Sie die Verriegelungspositionen des Buttons überprüfen, damit die Schaltfläche bei Größenveränderungen des Fensters immer in der rechten unteren Ecke bleibt.
   Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite.
   Klicken Sie auf die Symbole, bis oberes und linkes Symbol geöffnet sowie rechtes und unteres Symbol verriegelt sind.

Abb. 2.15 Locking für den ErledigtButton

Ihr Webseitenlayout sollte nun etwa so aussehen:

Abb. 2.16 Webseiten-Layout mit neuen Schaltflächen-Titeln



Abschnitt 8

TextField-Eigenschaften

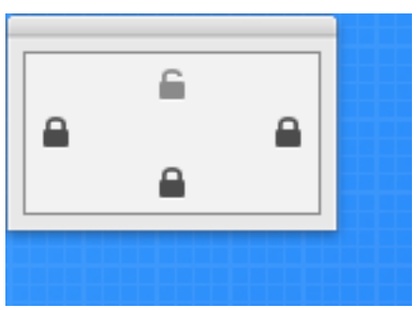
Im TextField wird der Benutzer die Aufgaben eingeben. Sie sollten die folgenden Properties verändern: **Name** und **Locking.**

1. Klicken Sie im Layout-Editor zunächst auf das TextField, um es auszuwählen. Der Inspector zeigt nun die WebTextField-Properties an.

2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „TextField1“ in „AufgabenFeld“. Drücken Sie *Return*, um die Veränderung in den Navigator zu übernehmen.

3. Nun sollten Sie das Locking ändern, damit das Text Field bei Größenveränderungen des Fensters mitskaliert wird.
   Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite.
   Klicken Sie auf die Symbole, bis das obere Symbol geöffnet ist und das linke, rechte und untere Symbol verriegelt sind.

Abb. 2.17 Locking für das AufgabenFeld



Projekttest

Das Projekt sichern

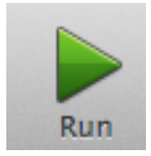
Sie sollten Ihr Projekt regelmäßig speichern – und grundsätzlich vor dem Ausführen.

1. Speichern Sie das Projekt via File ↠ Save.

2. Benennen Sie das Projekt „WebTutorial“ und klicken Sie auf Save.

Das Projekt ausführen

Jetzt, da Ihr Benutzerinterface-Layout vollständig ist, können Sie Ihr Programm testen. Klicken Sie auf das Run-Symbol in der Werkzeugleiste, um es ausführen zu lassen.



Wenn die Aufgabenverwaltung in Ihrem Browser angezeigt wird, können Sie mit den Schaltflächen interagieren, indem Sie auf sie klicken. Sie können Text in das TextField eingeben und das Fenster vergrößern und verkleinern und beobachten, wie sich die Locking-Einstellungen dabei auf die Elemente auswirken.

Allerdings verfügt die App über keinerlei Funktionalität. Wir müssen Programmcode hinzufügen, worum es im nächsten Kapitel gehen wird.

Schließen Sie den Browser-Tab oder das Fenster, um in die IDE zurückzugelangen.

Kapitel 3

Code hinzufügen

Der abschließende Schritt in der Erstellung eines Programms: Das Hinzufügen von Programmcode.

Die Hinzufügen-Schaltfläche

Code zum HinzufügenButton hinzufügen

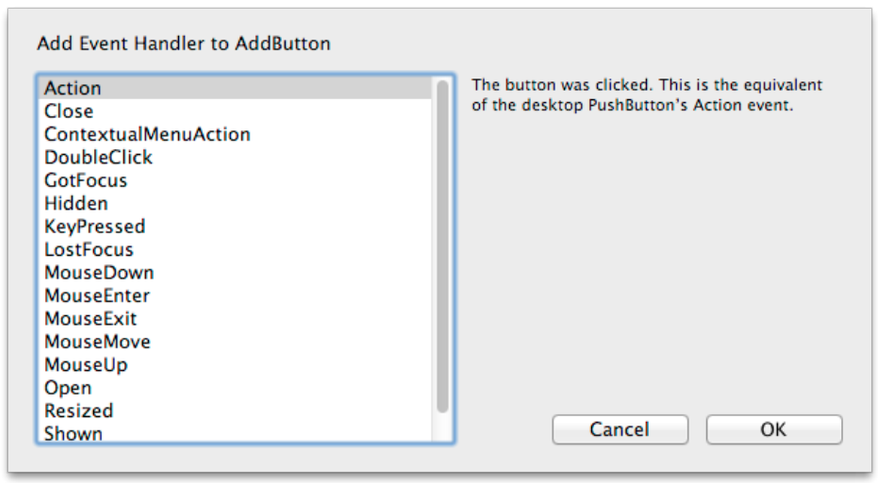
Der HinzufügenButton fügt Aufgaben zur Liste hinzu. Der zu erstellende Programmcode muss also den Text aus dem Ausgabenfeld nehmen und ihn als neue Zeile der Liste hinzufügen.

Ergänzen Sie den Code folgendermaßen:

1. Doppelklicken Sie in der Webseite auf den **HinzufügenButton** mit dem Titel „Hinzufügen“.

Der **Event-**Handler**-Hinzufügen-Dialog** (**Add Event Handler**) erscheint.

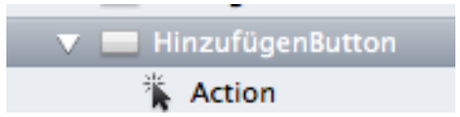
Abb. 3.1 Add Event Handler-Dialog



2. Wenn ein Benutzer auf einen WebButton klickt, wird der Programmcode im **Action-Event**-Handler ausgeführt. Sie wollen eine Aktion programmieren, die bei Klicken des Buttons ausgeführt wird. Wählen Sie daher Action aus der Event-Handler-Liste und klicken Sie auf OK.

Daraufhin wird der **Code-Editor** angezeigt. Im Navigator sehen Sie außerdem den Action-Event unterhalb des HinzufügenButtons.

Abb. 3.2 Action Event-Handler



3. Sie müssen also die Aufgabe bekommen, die in das AufgabenFeld eingegeben wurde. Eventuell denken Sie gerade, dass Sie den Text erhalten können, indem Sie den Namen des Felds, AufgabenFeld, adressieren.
   Das ist nah dran, trifft es aber nicht völlig.

   Stattdessen müssen Sie eine Property von Aufgabenfeld ansprechen.

Wann immer Sie auf eine Property eines Objekts zugreifen wollen, verwenden Sie den Namen des Objekts, gefolgt von einem Punkt und dahinter den Namen der Eigenschaft. Kurz gesagt: Verwenden Sie diese Syntax:

ObjektName.PropertyName

Dies nennt man Punktnotation (**dot notation**), eine häufig benutzte Schreibweise in objektorientierter Programmierung.

In diesem Fall heißt das Objekt **AufgabenFeld** und die gesuchte Property heißt **Text** (schauen Sie in die Sprachreferenz, um alle Properties eines Textfields kennenzulernen).

Die Syntax sieht also so aus:

```
AufgabenFeld.Text
```

4. Um eine Zeile zu einer Listnox hinzuzufügen, verwenden Sie die AddRow-Methode. Sie wissen bereits vom obigen Schritt, wie Sie die Texteigenschaft ansprechen. Verbinden Sie beides und Sie erhalten diesen Code:

   ```
   AufgabenListe.AddRow("", AufgabenFeld.Text)
   ```

   Wie Sie zuvor gesehen haben, besitzen Objekte Eigenschaften. Nun sehen Sie, dass Objekte ebenfalls **Methoden** besitzen können. **AddRow** ist eine der zahlreichen Methoden von Listboxen.

   Der obige Befehl fügt Werte in Form einer neuen Zeile an die ListBox an. Die erste Spalte enthält den „Erledigt“-Status und bleibt daher leer. Die zweite Spalte beinhaltet den Namen der Aufgabe.

5. Speichern Sie das Projekt durch Wahl von File ↠ Save.

6. Starten Sie die App und probieren Sie sie aus. Geben Sie Aufgaben in das Aufgabenfeld ein und klicken Sie auf „Hinzufügen“, um sie in der Liste auftauchen zu sehen.

7. Schließen Sie Browser-Tab oder -Fenster, um zum Code-Editor zurückzugelangen

Der ErledigtButton

Code zum ErledigtButton hinzufügen

Wenn der Benutzer auf diesen Knopf klickt, sollte die ausgewählte Aufgabe in der Liste mit einem Häkchen in der „Erledigt“-Spalte versehen werden (✓).

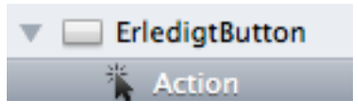
Gehen Sie dazu wie folgt vor:

1. Doppelklicken Sie auf der Webseite den **ErledigtButton** mit dem Titel „Erledigt“.

Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint. Wie erwähnt wird beim Klick auf einen WebButton der Action Event-Handler aufgerufen.
Sie wollen eine Aktion programmieren, die bei Klicken des Buttons ausgeführt wird. Wählen Sie daher Action aus der Event-Handler-Liste und klicken Sie auf OK.

Der Navigator stellt nun den Action-Event des ErledigtButtons dar und Sie gelangen in den Code-Editor des Handlers

Abb. 3.3 Action Event-Handler



2. Um eine Zeile zu verändern, müssen Sie zunächst wissen, welche Zeile überhaupt ausgewählt ist. In einer ListBox finden Sie die ausgewählte Zeile unter der Eigenschaft **ListIndex**.

3. Um den Wert einer bestimmten Zelle in einer ListBox zu ändern, verwenden Sie die **Cell**-Eigenschaft der Listbox, bei der Sie Zeile und Spalte der gewünschten Zelle angaben müssen. Der Code sieht wie folgt aus:

```
AufgabenListe.Cell(AufgabenListe.ListIndex, 0)= "✓"
```

   Dieser Befehl setzt das Zeichen für das Häkchen in spalte 0 der aktuell aktivierten Zeile.

4. Speichern Sie das Projekt.

5. Starten Sie die App und fügen Sie ein paar Aufgaben hinzu. Klicken Sie dann auf eine Aufgabe und daraufhin auf den „Erledigt“-Knopf. Ein Häkchen erscheint in Spalte 0 der angewählten Zeile.

6. Schließen Sie Browser-Tab oder -Fenster, um zum Code-Editor zurückzugelangen.

Der LöschButton

Code dem LöschButton hinzufügen

Mit dem LöschButton werden Aufgaben aus der Liste entfernt. Der von Ihnen zu schreibende Code muss die ausgewählte Zeile der Liste erkennen und diese aus ihr löschen können.

Gehen Sie dazu wie folgt vor:

1. Doppelklicken Sie im Fenster des Layout-Editors den **LöschButton**, den Sie „Löschen“ betitelt haben.

Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint. Wie bereits bei den anderen Schaltflächen wollen Sie den **Action**-Event-Handler verwenden, um Ihren Code ausführen zu lassen, wenn der Benutzer auf die Schaltfläche klickt. Wählen Sie also Action aus der Event-Handler-Liste und klicken Sie auf OK.

Im Navigator erscheint der Action-Event unterhalb des LöschButton-Steuerelements und der Code-Editor zeigt den (noch leeren) Code des Events an.

2. Sie müssen die ausgewählten Zeile erneut mithilfe der ListIndex-Eigenschaft herausfinden.

3. Verwenden Sie die Listbox-Methode **RemoveRow** , um eine Zeile aus einer Listbox zu entfernen. Sie übergeben RemoveRow die Zeilennummer der zu entfernenden Zeile als Parameter.
   Ihr Code sieht nun also so aus

```
AufgabenListe.RemoveRow(AufgabenListe.ListIndex)
```

4. Speichern Sie das Projekt durch Wahl von File ↠ Save.

5. Starten Sie die App und fügen Sie ein paar Aufgaben hinzu. Klicken Sie dann auf eine Aufgabe und daraufhin auf den LöschButton. Die Aufgabe verschwindet von der Liste.

6. Schließen Sie Browser-Tab oder -Fenster, um zum Code-Editor zurückzugelangen.

Debugging

Fehler finden und beseitigen

Obwohl Ihre App soweit ganz gut zu funktionieren scheint, lauern einige Fehler auf ihre Entdeckung. Als Bug bezeichnet man das Auftreten eines unvorhergesehenen Zustands in einer Anwendung, was oft zum Absturz führt. Haben Sie schon herausgefunden, was das Problem ist?

Ein Hinweis: Was wird passieren, wenn Sie den Erledigt- oder Löschnkopf drücken, ohne dass Sie eine Aufgabe angewählt haben? Versuchen Sie es mal!

1. Starten Sie die App und klicken Sie auf den „Erledigt“-Knopf, ohne dass Sie eine Aufgabe eingegeben haben.
   Versuchen Sie es! Ihre App wird in den Editor zurückschalten und eine Zeile optisch hervorheben. Ihr Programm ist durch eine **OutOfBounds**-Exception abgestürzt und Sie befinden sich jetzt im Debugger.

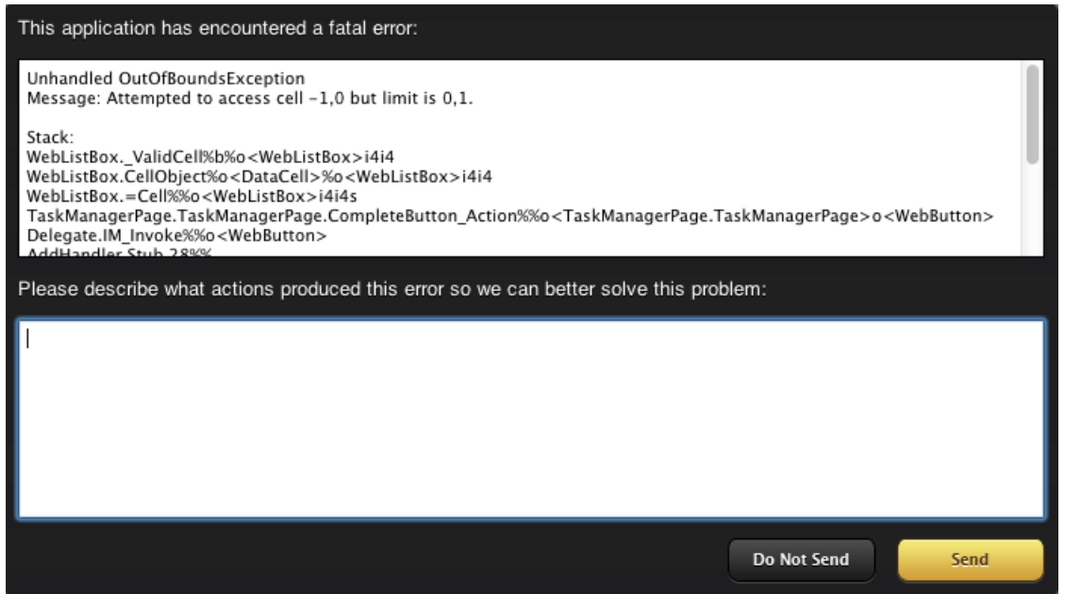
Abb. 3.4 Debugger-Hinweis an der Codezeile, die den Fehler verursacht hat

```
Sub Action()
    AufgabenListe.Cell(AufgabenListe.ListIndex, 0) = "✓"
End Sub
```

Der Fehler trat auf, weil Sie versuchten, eine Zeile zu entfernen (oder als erledigt zu markieren), die nicht existiert. Wenn keine Zeile in einer Listbox ausgewählt ist, liefert die Listindex-Property den Wert -1 zurück..Da dies eine existierende Zeile in der ListBox ist, liefert der Cell-Befehl eine Out of Bounds-Exception.

2. Klicken Sie auf den *Resume*-(Fortfahren)-Button in der Werkzeugleiste des Debuggers, um die Fehlermeldung in der Anwendung zu sehen.

Abb. 3.5 Ein Laufzeitfehler im Browser dargestellt



3. Schließen Sie Browser-Tab oder -Fenster, um zum Code-Editor zurückzugelangen.

Niemand will fehlerhaften Code. Erfreulicherweise ist dieser Fehler einfach zu beseitigen. Im Grunde müssen wir nur sicherstellen, dass eine existierende Zeile ausgewählt wurde, bevor die Lösch- oder Erledigt-Aktion ausgeführt wird.

1. Den nötigen Code dazu kennen Sie bereits:

   Dies ist der Code für den Action Event-Handler des LöschButtons:

   ```
   If AufgabenListe.ListIndex >= 0 Then
     AufgabenListe .RemoveRow(AufgabenListe.ListIndex)
   End If
   ```

2. Der Code für den ErledigtButton ist ganz ähnlich:

   ```
   If AufgabenListe.ListIndex >= 0 Then
     AufgabenListe.Cell(AufgabenListe.ListIndex, 0) = "✓"
   End If
   ```

3. In beiden Fällen stellt der Programmcode sicher, dass eine Zeile angewählt wurde, indem die ListIndex-Eigenschaft daraufhin abgefragt wird.

4. Speichern Sie das Projekt durch Wahl von File ↠ Save.

5. Starten Sie die App erneut und klicken Sie wieder den ErledigtButton ohne Anwahl einer Listenzeile an – kein Absturz mehr!

Kapitel 4

Die nächsten Schritte

Dachten Sie, wir wären fertig? Noch nicht ganz!

Abschnitt 1

Test der Aufgabenverwaltung

Unabdingbar: Der Programmtest

Sie sind nicht wirklich fertig mit dem Programmieren, nur weil Sie nun eine fertige Applikation erstellt haben. Ein guter Entwickler testet seine Anwendungen stets gründlich, um mögliche Probleme aufzuspüren.

Ein Problem haben Sie bereits gefunden und behoben – den Klick auf den Lösch- oder ErledigtButton ohne ausgewählte Listenzeile. Können Sie sich vorstellen, dass noch weitere Probleme existieren, die korrigiert werden sollten?

Starten Sie ihre App und spielen Sie ein wenig damit herum. Notieren Sie sich Dinge, die Sie gerne ändern möchten.

Im nächsten Abschnitt werden wir die Aufgabenerwaltung ein wenig verbessern.

Verbesserungen

Schaltflächenbedienbarkeit

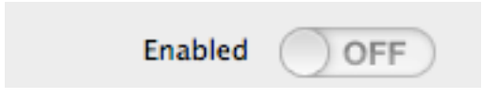
Ist Ihnen aufgefallen, dass einige Buttons in der Aufgabenverwaltung manchmal nicht verfügbar sein sollten? Der ErledigtButton beispielsweise sollte besser gar nicht klickbar sein, wenn keine Zeile ausgewählt ist. Momentan können Sie ihn anklicken, aber es passiert dann nichts. Und der HinzufügenButton sollte ebenso nur funktionieren, wenn auch eine Zeile in das Textfeld eingetragen wurde.

Es gibt verschiedene Methoden, dies zu erreichen. Eine davon besteht darin, die Schaltflächen zu deaktivieren, wenn sie nicht benutzbar sein sollen.

Gehen Sie dazu wie folgt vor:

1. Wählen Sie im Layout-Editor den **ErledigtButton** mit dem Titel „Erledigt“. Schalten Sie im Inspector die Property Enabled auf Off (aus).

Abb. 4.1 Die Enabled-Property im Inspector



2. Gehen Sie genauso mit dem **HinzufügenButton** um..

3. Stellen Sie ebenfalls die Enabled-Eigenschaft des **LöschButton** mit dem Titel „Löschen“ auf Off.

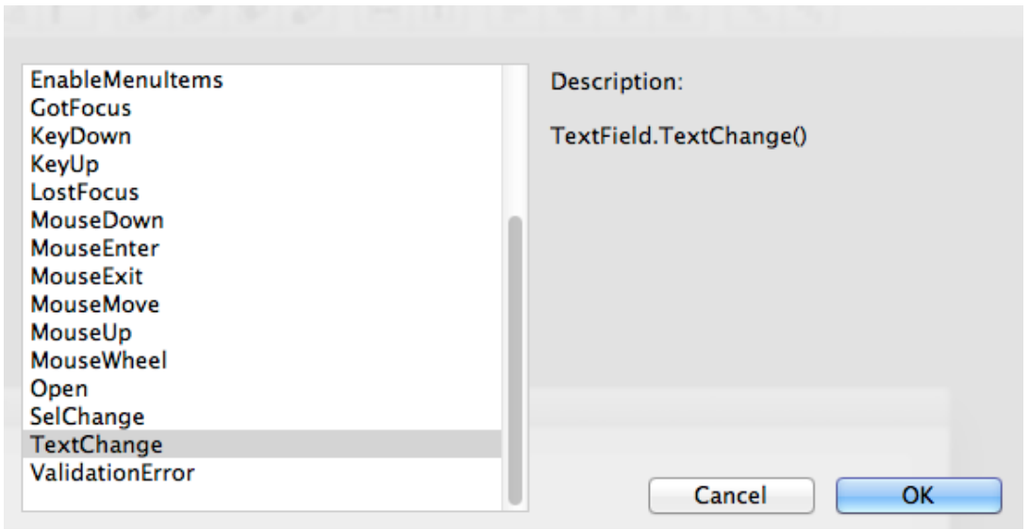
4. Nun fügen wir Programmcode hinzu, der den HinzufügenButton benutzbar macht, wenn Text im Aufgabenfeld steht.

   Doppelklicken Sie im Layout-Editor auf das **AufgabenFeld**-Steuerelement.

Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint, doch diesmal mit anderen verfügbaren Events. Jedes Steuerelement verfügt über einen klassentypischen Satz von Event-Handlern. Entsprechend sehen Sie nun die Event-Handler für ein TextField.
Wir wollen den HinzufügenButton deaktivieren, wenn kein Text im AufgabenFeld steht, und ihn verfügbar machen, wenn Text eingetragen ist. Der **TextChanged-Event** wird ausgelöst, wann immer der Text in einem TextField verändert wurde – sowohl durch den Benutzer als auch durch das Programm selbst. Wählen Sie ihn also aus.

DerNavigator zeigt nun den TextChanged-Event unterhalb das AufgabenFeld-Steuerelements an.

Abb. 4.2 Event-Handler für ein TextField



5. Geben Sie diesen Code ein:

```
If Me.Text <> "" Then
  HinzufügenButton.Enabled = True
Else
  HinzufügenButton.Enabled = False
End If
```

Beachten Sie bitte, dass wir hier **Me.Text** verwenden, um den Text des TextFields zu erhalten. Wir hätten die Eigenschaft ebenso über Aufgabenfeld.Text ansprechen können. Warum also Me?

Me verweist auf das aufrufende Objekt selbst ohne individuellen Namen – Sie können somit den Namen des Steuerelements ändern oder Code erstellen, der in allen Instanzen eigener Klassen funktioniert,egal welchen Namen diese tragen.

Dieser Code überprüft die Eigenschaft des TestFields daraufhin, ob etwas in ihr steht.Falls ja, wird der HinzufügenButton aktivierbar geschaltet, indem seine Property Enabled auf **True** (Wahr) gesetzt wird. Andernfalls wird Enabled auf **False** (Falsch) gesetzt

6. In Kapitel 3 Abschnitt 4 haben Sie bereits Code zu den Lösch- und ErledigtButtons hinzugefügt, um zu verhindern, dass ihre Aktionen ausgelöst werden, wenn keine Zeile in der ListBox angewählt ist. Jetzt können Sie ebenfalls dafür sorgen, dass die Schaltflächen gar nicht benutzbar sind, wenn die obige Bedingung nicht erfüllt wurde. Dazu benötigen Sie wieder die Eigenschaft ListIndex der ListBox.

7. Doppelklicken Sie auf das AufgabenListe-Steuerelement. Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint, diesmal mit den verfügbaren Event Handlers für eine WebListbox. Der **SelectionChanged**-Event wird aufgerufen, wann immer sich die Auswahl in einer WebListBox verändert. Wählen Sie SelectionChanged und klicken Sie auf OK.

8. Geben Sie diesen Code ein:

```
If Me.ListIndex >= 0 Then
  LöschButton.Enabled = True
  ErledigtButton.Enabled = True
Else
  LöschButton.Enabled = False
  ErledigtButton.Enabled = False
```

9. Speichern Sie das Projekt durch Wahl von File ↠ Save.

10. Starten Sie die App und testen Sie sie. Wie Sie sehen, ist der HinzufügenButton nun beim Start deaktiviert. Sobald Sie Text in das AufgabenFeld eingeben, wird der Knopf anwählbar. Und wenn Sie den Text wieder löschen, wird der Knopf wieder grau dargestellt.

Ebenso verändert sich die Verfügbarkeit von Lösch- und ErledigtButtons bei Klicks auf die ListBox.

Weitere Verbesserungen

Spielen Sie ein wenig weiter mit Ihrer App. Sie finden bestimmt Punkte, die Sie weiter verbessern möchten. Soll die Anwendung für mehrere Benutzer mit individuellen Aufgabenlisten verfügbar sein? Soll es gemeinsame Aufgaben geben, die geteilt werden können?
Für eine Vielzahl von Programmiertechniken finden Sie Anregungen in den Beispielprojekten, die in Ihrem Xojo-Verzeichnis installiert wurden.

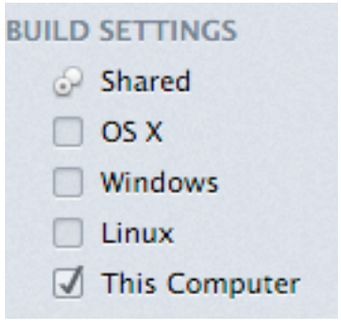
Deployment (Veröffentlichung) einer Web-App

Teilen Ihrer Anwendung

Da Sie jetzt diese feine Anwendung erstellt haben, möchten Sie sie wahrscheinlich mit der Welt teilen. Sie können die Web-App zurzeit lokal in Ihrem Browser ausführen, aber normalerweise sollte sie auf einem Webserver laufen und über Browserverbindungen ansprechbar sein.
Um Ihre Anwendung so verfügbar zu machen (es hat sich hier der Begriff **Deployment** (so viel wie Auslieferung) eingebürgert), müssen Sie einen **Build** erzeugen und ihn dann auf einen Webserver deployen.

Ihre Web-App kann für jede unterstützte Plattform erzeugt werden: Mac OS X, Windows und Linux. Die meisten Webserver verwenden Linux, weshalb wir die App in diesem Beispiel auch dafür erzeugen. Klicken Sie also in den Build-Settings auf „Linux“.

Abb. 4.3 Build-Einstellungen



Deployment-Optionen

Als nächstes müssen Sie entscheiden, ob Sie eine eigenständig lauffähige App (Stand-Alone) oder eine CGI-Web-App erzeugen wollen.

Stand-Alone-Web-App

Eine Stand-Alone-Web-App ist eine Anwendung, die Sie manuell auf Ihrem Webserver starten müssen – üblicherweise aus dem Kommandozeileninterface heraus. Sie muss dann kontinuierlich auf dem Webserver laufen, um ansprechbar zu sein.
Eine Stand-Alone-Web-App wird über einen Port angesprochen, den Sie vor dem Build-Prozess angeben müssen.
Eine solche App besteht im Grunde aus der Kombination eines Webservers mit Ihrem kompilierten Programmcode.

Stand-Alone-Web-Apps können WebSockets verwenden, ein Feature, das die Ausführungsgeschwindigkeit von Web-Apps durch die Bereitstellung schnellerer bidirektionaler Kommunikationskanäle erhöht.

Eine deployte Stand-Alone-Web-App wird über ihre URL angesprochen, wie etwa:
http://www.mywebsite.com:8080

CGI-Web-App

Eine Web-App, die **CGI** verwendet, benötigt entweder einen Apache- oder einen IIS (Microsoft Internet Information Services)-Webserver. Dieser Server kommuniziert dann via CGI mit Ihrer App. Ein Perl-Script, das beim Build einer CGI-Web-App erzeugt wird, kümmert sich um den Datenaustausch zwischen Webserver und Web-App.

Momentan können CGI-Apps keine WebSockets verwenden. Einige Browser, insbesondere Safari, zeigen den Ladebalken auch dann noch an, wenn die Seite schon längst geladen ist. Die vom Webserver verwendete Kommunikationsmethode mit der Web-App bringt diese Probleme hervor.

Da ein CGI-Deployment Ihren bereits bestehenden Webserver verwendet, müssen Sie keinen Port angeben, wenn Sie Ihre App ansprechen wollen. Eine typische URL für eine CGI-App hat daher diese Form:

http://www.mywebsite.com/cgi-bin/mywebapp.cgi

Deployment

Aufgrund der großen Vielzahl an Servern und ihren individuellen Konfigurationen liegt eine detaillierte Darstellung des Deployments leider jenseits der Möglichkeiten dieses Tutorials.

Im Allgemeinen besteht das Deployment einer Web-App auf einen Linux-Server aus den folgenden Schritten:

1. **Kompilieren** Sie Ihre Web-App für Linux.
2. Verbinden Sie sich via FTP mit Ihrem Webserver.
3. Laden Sie Ihre Web-App (einschließlich des „Libs“-Verzeichnisses) auf Ihren Server hoch
4. Stellen Sie sicher, dass das „Execute“-Bit in den Zugriffsrechten der hochgeladenen Dateien immer noch gesetzt ist. Einige FTP-Programme verändern dieses während des Uploads eigenmächtig.

Die Details können sehr viel komplexer sein. In unserer Online-Dokumentation finden Sie weitere Hinweise.

Die Xojo Cloud

Sofern Sie das Desktop-Tutorial schon durchgearbeitet haben, werden Sie festgestellt haben, dass die Erzeugung einer Stand-Alone-Desktop-App mit einem Klick möglich ist – und die Weitergabe kann dann auf jedem erdenkbaren Datenweg erfolgen.

Den selben Komfort genießen Sie beim Deployment von Web-Apps mit der Xojo Cloud:
Gegen eine monatliche Gebühr wird Ihnen Speicherplatz auf einem vorkonfigurierten Linux-Server mit neuesten Sicherheitstechnologien reserviert – Installation, Wartung und Update des Servers entfallen damit.

Durch einen Klick auf den „Deploy“-Button in der Werkzeugleiste von Xojo wird Ihre Web-App kompiliert, auf den Server hochgeladen und bereitgestellt. FTP-Handling, Überprüfen von Zugriffsrechten und weitere Konfigurationen entfallen somit ebenso.

Weitere Informationen zur Xojo Cloud, dem dahinterstehenden Sicherheitssystem und den verfügbaren Serverzentren finden Sie in der Xojo Cloud-FAQ.

Abb. 4.4 Der Deploy-Button für die Xojo Cloud

Abschnitt 4

Fertig!

Herzlichen Glückwunsch!

Sie haben das Web-Tutorial erfolgreich absolviert und eine voll funktionsfähige Anwendung erstellt.

Als nächsten Schritt auf Ihrer Xojo-Entdeckungsreise empfehlen wir Ihnen die Lektüre des Benutzerhandbuchs, das Xojo in seiner Gänze abdeckt.

Ziehen Sie ebenfalls die Sprachreferenz zurate, die die Details der Sprachelemente, Klassen und anderer Xojo-Features schnell nachschlagbar auflistet.